

09-21-00

A

Express Mail Label No. EK799875474US

UTILITY PATENT APPLICATION TRANSMITTAL (Small Entity)

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Docket No.
CREO.005US0Total Pages in this Submission
42

TO THE ASSISTANT COMMISSIONER FOR PATENTS

Box Patent Application
Washington, D.C. 20231

Transmitted herewith for filing under 35 U.S.C. 111(a) and 37 C.F.R. 1.53(b) is a new utility patent application for an invention entitled:

METHOD AND APPARATUS FOR REMOTELY DEBUGGING AN APPLICATION PROGRAM OVER THE INTERNET

and invented by:

Lee, Kuo-Chun; and Chen, Tsung-Yen (Eric)

If a **CONTINUATION APPLICATION**, check appropriate box and supply the requisite information:☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: _____

Which is a:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: _____

Which is a:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: _____

Enclosed are:

Application Elements

1. ☒ Filing fee as calculated and transmitted as described below
2. ☒ Specification having 25 pages and including the following:
 - a. ☒ Descriptive Title of the Invention
 - b. ☐ Cross References to Related Applications (if applicable)
 - c. ☐ Statement Regarding Federally-sponsored Research/Development (if applicable)
 - d. ☐ Reference to Microfiche Appendix (if applicable)
 - e. ☒ Background of the Invention
 - f. ☒ Brief Summary of the Invention
 - g. ☒ Brief Description of the Drawings (if drawings filed)
 - h. ☒ Detailed Description
 - i. ☒ Claim(s) as Classified Below
 - j. ☒ Abstract of the Disclosure

UTILITY PATENT APPLICATION TRANSMITTAL
(Small Entity)

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Docket No.
CREO.005US0

Total Pages in this Submission
42

Application Elements (Continued)

3. ☒ Drawing(s) *(when necessary as prescribed by 35 USC 113)*
a. ☒ Formal b. ☐ Informal Number of Sheets 5
4. ☒ Oath or Declaration
a. ☒ Newly executed *(original or copy)* ☐ Unexecuted
b. ☐ Copy from a prior application (37 CFR 1.63(d)) *(for continuation/divisional application only)*
c. ☒ With Power of Attorney ☐ Without Power of Attorney
d. ☐ DELETION OF INVENTOR(S)
Signed statement attached deleting inventor(s) named in the prior application,
see 37 C.F.R. 1.63(d)(2) and 1.33(b).
5. ☐ Incorporation By Reference *(usable if Box 4b is checked)*
The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under
Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby
incorporated by reference therein.
6. ☐ Computer Program in Microfiche
7. ☐ Genetic Sequence Submission *(if applicable, all must be included)*
a. ☐ Paper Copy
b. ☐ Computer Readable Copy
c. ☐ Statement Verifying Identical Paper and Computer Readable Copy

Accompanying Application Parts

8. ☒ Assignment Papers *(cover sheet & documents)*
9. ☐ 37 CFR 3.73(b) Statement *(when there is an assignee)*
10. ☐ English Translation Document *(if applicable)*
11. ☐ Information Disclosure Statement/PTO-1449 ☐ Copies of IDS Citations
12. ☐ Preliminary Amendment
13. ☒ Acknowledgment postcard
14. ☒ Certificate of Mailing
☐ First Class ☒ Express Mail *(Specify Label No.):* EK799875474US

UTILITY PATENT APPLICATION TRANSMITTAL
(Small Entity)

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Docket No.
CREO.005US0

Total Pages in this Submission
42

Accompanying Application Parts (Continued)

15. ☐ Certified Copy of Priority Document(s) (if foreign priority is claimed)
16. ☒ Small Entity Statement(s) - Specify Number of Statements Submitted: 1
17. ☐ Additional Enclosures (please identify below):

Fee Calculation and Transmittal

CLAIMS AS FILED

For	#Filed	#Allowed	#Extra	Rate	Fee
Total Claims	37	- 20 =	17	x \$9.00	\$153.00
Indep. Claims	8	- 3 =	5	x \$39.00	\$195.00
Multiple Dependent Claims (check if applicable) <input type="checkbox"/>					\$0.00
BASIC FEE					\$345.00
OTHER FEE (specify purpose) _____					\$0.00
TOTAL FILING FEE					\$693.00

- ☒ A check in the amount of **\$693.00** to cover the filing fee is enclosed.
- ☐ The Commissioner is hereby authorized to charge and credit Deposit Account No. _____ as described below. A duplicate copy of this sheet is enclosed.
- ☐ Charge the amount of _____ as filing fee.
- ☐ Credit any overpayment.
- ☐ Charge any additional filing fees required under 37 C.F.R. 1.16 and 1.17.
- ☐ Charge the issue fee set in 37 C.F.R. 1.18 at the mailing of the Notice of Allowance, pursuant to 37 C.F.R. 1.311(b).

Dated: September 19, 2000

Chitra H. Okumoto
Signature

cc:

CERTIFICATE OF MAILING BY "EXPRESS MAIL" (37 CFR 1.10)Applicant(s): **Lee, Kuo-Chun et al.**

Docket No.

CREO.005US0

Serial No.

Filing Date

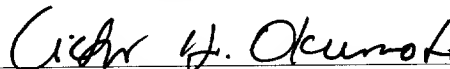
Examiner

Group Art Unit

Invention:

METHOD AND APPARATUS FOR REMOTELY DEBUGGING AN APPLICATION PROGRAM OVER THE INTERNET1c784 U.S. PTO
09/664941
09/19/00I hereby certify that this **APPLICATION FOR UNITED STATES PATENT***(Identify type of correspondence)*

is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under
37 CFR 1.10 in an envelope addressed to: The Assistant Commissioner for Patents, Washington, D.C. 20231 on
September 19, 2000

*(Date)***Victor H. Okumoto***(Typed or Printed Name of Person Mailing Correspondence)**(Signature of Person Mailing Correspondence)***EK799875474US***("Express Mail" Mailing Label Number)***Note: Each paper must have its own certificate of mailing.**

**VERIFIED STATEMENT (DECLARATION) CLAIMING SMALL ENTITY
STATUS (37 CFR 1.9(f) AND 1.27 (c)) - SMALL BUSINESS CONCERN**

Docket No.
CREO.005US0

Serial No.

Filing Date

Patent No.

Issue Date

Applicant/

Patentee: Lee,Kuo-Chun; and Chen,Tsung-Yen (Eric)

Invention:

**METHOD AND APPARATUS FOR REMOTELY DEBUGGING AN APPLICATION PROGRAM OVER THE
INTERNET**

I hereby declare that I am:

- ☐ the owner of the small business concern identified below:
- ☒ an official of the small business concern empowered to act on behalf of the concern identified below:

NAME OF CONCERN: CreOsys Inc.ADDRESS OF CONCERN: 39560 Stevenson Pl., Suite 221, Fremont CA 94539

I hereby declare that the above-identified small business concern qualifies as a small business concern as defined in 37 CFR 121.3-18, and reproduced in 37 CFR 1.9(d), for purposes of paying reduced fees under Section 41(a) and (b) of Title 35, United States Code, in that the number of employees of the concern, including those of its affiliates, does not exceed 500 persons. For purposes of this statement, (1) the number of employees of the business concern is the average over the previous fiscal year of the concern of the persons employed on a full-time, part-time or temporary basis during each of the pay periods of the fiscal year, and (2) concerns are affiliates of each other when either, directly or indirectly, one concern controls or has the power to control the other, or a third party or parties controls or has the power to control both.

I hereby declare that rights under contract or law have been conveyed to and remain with the small business concern identified above with regard to the above identified invention described in:

- ☒ the specification filed herewith with title as listed above.
- ☐ the application identified above.
- ☐ the patent identified above.

If the rights held by the above-identified small business concern are not exclusive, each individual, concern or organization having rights to the invention is listed on the next page and no rights to the invention are held by any person, other than the inventor, who could not qualify as an independent inventor under 37 CFR 1.9(c) or by any concern which would not qualify as a small business concern under 37 CFR 1.9(d) or a nonprofit organization under 37 CFR 1.9(e).

Each person, concern or organization to which I have assigned, granted, conveyed, or licensed or am under an obligation under contract or law to assign, grant, convey, or license any rights in the invention is listed below:

- ☒ no such person, concern or organization exists.
☐ each such person, concern or organization is listed below.

FULL NAME _____
 ADDRESS _____

☐ Individual ☐ Small Business Concern ☐ Nonprofit Organization

FULL NAME _____
 ADDRESS _____

☐ Individual ☐ Small Business Concern ☐ Nonprofit Organization

FULL NAME _____
 ADDRESS _____

☐ Individual ☐ Small Business Concern ☐ Nonprofit Organization

FULL NAME _____
 ADDRESS _____

☐ Individual ☐ Small Business Concern ☐ Nonprofit Organization

Separate verified statements are required from each named person, concern or organization having rights to the invention averring to their status as small entities. (37 CFR 1.27)

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

NAME OF PERSON SIGNING: Kuo-Chun Lee

TITLE OF PERSON SIGNING _____

OTHER THAN OWNER: President

ADDRESS OF PERSON SIGNING: CreOsyst Inc.

39560 Stevenson Pl., Suite 221

Fremont, CA 94539

SIGNATURE:  DATE: 9/19/2000

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
APPLICATION FOR PATENT

**METHOD AND APPARATUS FOR REMOTELY DEBUGGING AN
APPLICATION PROGRAM OVER THE INTERNET**

Inventors: Kuo-Chun Lee, and
Tsung-Yen (Eric) Chen,

FIELD OF THE INVENTION

The present invention generally relates to methods for software vendors to provide technical assistance to their customers and in particular, to a method and apparatus for remotely debugging an application program over the Internet.

BACKGROUND OF THE INVENTION

It is common business practice for customers to purchase application programs or software over the Internet by downloading them from vendor web sites. This method of purchasing software is both convenient for customers, and cost effective for vendors. Customers do not waste time or gas traveling to and looking for retail stores stocking the software, and vendors eliminate middleman expenses and avoid packaging and media costs.

Unfortunately, when customers experience problems using the purchased software, the prevalent method of debugging those problems continues to be the old fashioned method of calling in for customer telephone assistance. This approach results too often in long waits to talk to technical support personnel, and limited diagnostic assistance once such personnel become available since it

may be difficult for them to replicate the customer's problem on their computers.

OBJECTS AND SUMMARY OF THE INVENTION

Accordingly, it is an object of the present invention to provide an improved method and apparatus for software vendors to provide technical assistance to their customers.

Another object is to provide a method and apparatus for software vendors to provide technical assistance to their customers over the Internet.

Still other objects are to provide a method and apparatus for remotely debugging an application program over the Internet.

These and additional objects are accomplished by the various aspects of the present invention, wherein briefly stated, one aspect is a method implemented in a client computer for remotely debugging an application program over the Internet, comprising: (a), establishing a connection between the client computer and a server computer over the Internet; (b), receiving a request from a debug program of the server computer; (c), causing an application program of the client computer to generate a response to the request; and (d), transmitting an indication of the response back to the debug program; and (e), repeating (b), (c) and (d) multiple times so as to run the application program through a diagnostic sequence.

In another aspect, an apparatus for remotely debugging an application program over the Internet, comprises a client computer having an interface program

for: (a), establishing a connection between the client computer and a server computer over the Internet; (b), receiving a request from a debug program of the server computer; (c), causing an application program of the client computer to generate a response to the request; and (d), transmitting an indication of the response back to the debug program; and (e), repeating (b), (c) and (d) multiple times so as to run the application program through a diagnostic sequence.

In another aspect, a method implemented in a client computer for remotely debugging an application program over the Internet, comprises: (a), establishing a connection between the client computer and a server computer over the Internet; (b), receiving a request from a debug program of the server computer; (c), causing an application program of the client computer to respond to the request; (d), generating a graphics file including pixel information for a graphics image displayed on a display screen of the client computer; and (e), transmitting the graphics file to the server computer so that the graphics image is displayable on a display screen of the server computer.

In another aspect, an apparatus for remotely debugging an application program over the Internet, comprises a client computer having an interface program for: (a), establishing a connection between the client computer and a server computer over the Internet; (b), receiving a request from a debug program of the server computer; (c), causing an application program of the client computer to respond to the request; (d), generating a graphics file including pixel information for a graphics

image displayed on a display screen of the client computer; and (e), transmitting the graphics file to the server computer so that the graphics image is displayable on a display screen of the server computer.

In another aspect, a method implemented in a server computer for remotely debugging an application program over the Internet, comprises: (a), receiving a request from a client computer over the Internet to debug an application program of the client computer; (b), transmitting back to the client computer a request for the application program to take an action; (c), receiving an indication of a response of the application program action back from the client computer; and (d), repeating (b) and (c) multiple times so as to run the application program through a diagnostic sequence.

In another aspect, an apparatus for remotely debugging an application program over the Internet, comprises a server computer having a debug program for: (a), receiving a request from a client computer over the Internet to debug an application program of the client computer; (b), transmitting back to the client computer a request for the application program to take an action; (c), receiving an indication of a response of the application program action back from the client computer; and (d), repeating (b) and (c) multiple times so as to run the application program through a diagnostic sequence.

In yet another aspect, a method implemented in a server computer for remotely debugging an application program over the Internet, comprises: (a), receiving a request from a client computer over the Internet to debug an application program of the client computer; (b),

transmitting back to the client computer a request for the application program to take an action; (c), receiving a graphics file including pixel information for a graphics image displayed on a display screen of the client computer in response to the action; (d), displaying the graphics image on a display screen of a server computer; and (e), repeating (b) through (d) multiple times so as to allow a user of the server computer to interactively debug the application program by transmitting requests for the application program to take certain actions in consideration of graphics images defined in graphics files received from the client computer in response to prior such requests.

In still another aspect, an apparatus for remotely debugging an application program over the Internet, comprises a server computer having a debug program for: (a), receiving a request from a client computer over the Internet to debug an application program of the client computer; (b), transmitting back to the client computer a request for the application program to take an action; (c), receiving a graphics file including pixel information for a graphics image displayed on a display screen of the client computer in response to the action; (d), displaying the graphics image on a display screen of a server computer; and (e), repeating (b) through (d) multiple times so as to allow a user of the server computer to interactively debug the application program by transmitting requests for the application program to take certain actions in consideration of graphics images defined in graphics files received from the client computer in response to prior such requests.

Additional objects, features and advantages of the various aspects of the present invention will become apparent from the following description of its preferred embodiments, which description should be taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 illustrates a diagram of a system for remotely debugging an application program over the Internet.

FIG. 2 illustrates a client display screen under control of an application program.

FIG. 3 illustrates a client display screen under control of an interface program.

FIG. 4 illustrates a flow diagram of a method implemented in a client computer for remotely debugging an application program over the Internet.

FIG. 5 illustrates a flow diagram of a method implemented in a server computer for remotely debugging an application program over the Internet.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. 1 illustrates a diagram of a system **100** for remotely debugging an application program **12** over the Internet **20**. Included in the system **100** are a client computer **10** having an application program **12** and an interface program **14** residing on it, and a server computer **30** having a debug program **32** residing on it. A customer of the application program **12** controls the client computer **10**, and a vendor of the application program **12** controls the

server computer 30. The client computer 10 and the server computer 30 are coupled through the Internet 20. The debug program 32 facilitates preprogrammed and vendor debugging of the application program 12.

The interface program 14 controls the application program 12 and communicates with the debug program 32 during the debug mode. When the debug program 32 transmits a request to be acted upon by the application program 12, the interface program 14 receives the request and causes the application program 12 to generate a response to the request. The interface program 14 then transmits an indication of the response back to the debug program 32.

In one form of indication, the interface program 14 simply returns the response to the debug program 32. In another form of indication, the interface program 14 captures pixel information for a graphics image on the display screen of the client computer 10 from a frame buffer associated with the client computer 10, converts the pixel information into a graphics file of a selected graphics file format, and transmits the graphics file to the debug program 32. The graphics file format may be any one of a number of standard formats such as JPEG, GIF or TIF. This form of indication allows the vendor operator to see what is being displayed on the display screen of the client computer 10. This is especially useful when the vendor operator is unable to replicate the response generated by the application program 12 running on the client computer 10 with a corresponding response generated by a copy of the application program 12 running on the server computer 30.

As used herein, the purchaser of the application program **12** is referred to as the "customer", the operator or user of the client computer **10** is referred to as the "customer operator", the vendor company of the application program **12** is referred to as the "vendor", and the operator or user of the server computer **30** is referred to as the "vendor operator". Vendor operators are typically customer technical support personnel. The interface program **14** is typically sold or licensed by the vendor along with the application program **12**.

FIG. 2 illustrates a simplified example of the client display screen **200** under the control of the application program **12**. A tool bar area **201** includes numerous buttons (not shown) for typical customer operator control input, and a debug button **202**. The interface program **14** is activated when the customer operator clicks the debug button **202**. Upon its activation, the interface program **14** takes control of the application program **12**, establishes a connection with the debug program **32**, receives requests from the debug program **32**, and transmits responses or indications of responses back to the debug program **32**. A client work area **203** is also shown on the client display screen **200**. The tool bar area **201** and the client work area **203** are legacies of the application program **12**, whereas the debug button **202** facilitates an added feature for remotely debugging the application program **12** over the Internet **20**.

FIG. 3 illustrates a simplified example of the client display screen **200** under the control of the interface program **14**. The tool bar area **201** is generally no longer responsive to customer operator input, except for

the debug button **202**. When the customer operator clicks the debug button **202** in this case, the interface program **14** is deactivated and control of the application program **12** returns to the customer operator. Superimposed over the client work area **203** is an exemplary window **304** used for communications between the customer operator and the debug program **32** or a vendor operator.

In one use of the window **304**, the interface program **14** prompts the customer operator for a user identification and password. After the customer operator provides the requested information, the interface program **14** transmits the user identification and password to the debug program **32** for verification. In another use of the window **304**, the interface program **14** presents a check-the-box type questionnaire to the customer operator to facilitate efficient diagnosis of the problem to be debugged. After the customer operator clicks the appropriate boxes, the interface program **14** transmits the information to the debug program **32**, which in turn, initiates appropriate diagnostic sequences corresponding to each checked box. In still another use of the window **304**, an instant messaging or chat room type area facilitates communication back and forth between the customer operator and the vendor operator to further facilitate diagnosis of the problem to be debugged.

FIG. 4 illustrates a flow diagram of a method implemented in a client computer **10** for remotely debugging an application program **12** over the Internet **20**. In **401**, the interface program **14** detects a debug request initiated by the customer operator clicking the debug button **202**, and as a result, becomes activated. After becoming activated,

the interface program **14** takes control of the application program **12** and establishes a connection with the debug program **32** of the server computer **30** over the Internet **20**. The interface program **14** knows the Internet address of the debug program **32** in this case, because it has been preprogrammed into the interface program **12** by the vendor.

In **402**, the interface program **14** transmits identification information to the debug program **32** of the server computer **30** over the Internet **20**. A first set of identification information includes identification of the application program **12**, and identification of the client computer **10**. This information allows the debug program **32** to verify that the application program **12** is still under warranty or maintenance contract, and that the client computer **10** is authorized to run the application program **12**. The identifications of the application program and client computer **10** may be sent automatically by the interface program **14** according to its programming after establishing connection with the debug program **32**, or sent by the interface program **14** after the debug program **32** has made a request for such information to the interface program **14**. A second set of identification information includes a user identification and a password. The customer operator provides this information in a conventional fashion after being prompted by the interface program **14**. Providing the user identification and password allows the debug program **32** to verify that the customer operator is authorized to contact the vendor company for debug support. If verification fails for either the first or second set of identification information, the interface program **14** displays a failure message received from the

debug program **32** in the window **304**. The interface program **14** is then deactivated, and control of the application program **12** is passed back to the customer operator.

In **403**, the interface program **14** receives a request for preliminary information from the debug program **32**. The preliminary information request may take the form of a fill-in-the-box questionnaire preprogrammed in and automatically transmitted by the debug program **32**, or instant messaging or chat room type questions from the vendor operator. In either case, the questionnaire or questions are displayed in the window **304**. The customer operator then responds to the questionnaire by checking the appropriate boxes, or answers the instant messaging or chat room type questions, as the case may be, and the interface program **14** transmits the information back to the debug program **32**. The purpose of such preliminary information is to make the debug process more efficient, such as narrowing the choices of diagnostic sequences to take in subsequent debugging activities.

In **404** through **406**, the interface program **14** takes certain actions in response to a diagnostic sequence received from the debug program **32**. The diagnostic sequence may be preprogrammed into the debug program **32**, or it may be provided to the debug program **32** from the vendor operator. It may also be a combination of a preprogrammed diagnostic sequence and vendor operator provided diagnostic sequence. In this case, the diagnostic sequence typically starts out as a preprogrammed diagnostic sequence that is paused at some point by the vendor operator. The vendor operator then takes over control of the debug program **32** at this point, and generates his or her own diagnostic

sequence so as to interactively debug the application program **12**.

In **404**, the interface program **14** receives a request from the debug program **32** to be redirected to the application program **12**. As previously described, the request is one of a diagnostic sequence directed to debug the application program **12**. In **405**, the interface program **14** causes the application program **12** to generate a response to the request. In **406**, the interface program **14** transmits an indication of the response back to the debug program **32**. As previously described, the indication may be the response itself, or a graphics file including pixel information for a graphics image displayed on the display screen of the client computer **10**. Then **404** through **406** are repeated multiple times so as to run the application program **12** through the diagnostic sequence.

FIG. 5 illustrates a flow diagram of a method implemented in a server computer **30** for remotely debugging an application program **12** over the Internet **20**. In **501**, the debug program **32** establishes a connection with the interface program **14** of the client computer **10**, as further described in reference to **401**. In **502**, the debug program **32** receives identification information from the interface program **14**, and verifies such information, as further described in reference to **402**. In **503**, the debug program **32** transmits a preliminary information request to the interface program **14**, and receives customer operator responses back, as further described in reference to **403**. By performing **504** through **506** multiple times, the debug program **32** or vendor operator generates and provides the diagnostic sequence to the interface program **14** in order to

debug the application program **12**, as further described in reference to **404** through **406**. In particular, in **504**, the debug program **32** transmits a diagnostic sequence request for the application program **12** to the interface program **14**. In **505**, the debug program **32** receives an indication of the response of the application program **12** to the diagnostic sequence request back from the interface program **14**. In **506**, the debug program **32** analyzes the response and determines a next diagnostic sequence request for the application program **12** according to a preprogrammed diagnostic sequence, or the vendor operator analyzes the response and determines a next diagnostic sequence request for the application program **12** according to a vendor operator determined diagnostic sequence.

Although the various aspects of the present invention have been described with respect to a preferred embodiment, it will be understood that the invention is entitled to full protection within the full scope of the appended claims.

CLAIMS

We claim:

1. A method implemented in a client computer for remotely debugging an application program over the Internet, comprising:

- (a) establishing a connection between a client computer and a server computer over the Internet;
- (b) receiving a request from a debug program of said server computer;
- (c) causing an application program of said client computer to generate a response to said request; and
- (d) transmitting an indication of said response back to said debug program; and
- (e) repeating (b), (c) and (d) multiple times so as to run said application program through a diagnostic sequence.

2. The method according to claim 1, wherein said diagnostic sequence is provided to said debug program by a user of said server computer.

3. The method according to claim 1, wherein said diagnostic sequence is preprogrammed into said debug program.

4. The method according to claim 3, further comprising detecting a debug request initiated by a user of said client computer, prior to (a).

5. The method according to claim 4, further comprising transmitting identifications of said application program and said client computer to said server computer, after (a).

6. The method according to claim 4, further comprising transmitting a user identification and a password provided by a user of said client computer to said server computer, after (a).

7. The method according to claim 3, wherein the running of said preprogrammed diagnostic sequence is paused by a user of said server computer and control of said debug program is transferred to said user of said server computer, and further comprising:

(f) receiving a request initiated by said user of said server computer;

(g) causing said application program to respond to said request;

(h) generating a graphics file including pixel information for a graphics image displayed on a display screen of said client computer; and

(i) transmitting said graphics file to said server computer so that said graphics image is displayed on a display screen of said server computer.

8. The method according to claim 7, further comprising repeating (f) through (i) multiple times so as to allow said user of said server computer to remotely debug said application program.

9. An apparatus for remotely debugging an application program over the Internet, comprising a client computer including an interface program for:

- (a) establishing a connection between a client computer and a server computer over the Internet;
- (b) receiving a request from a debug program of said server computer;
- (c) causing an application program of said client computer to generate a response to said request; and
- (d) transmitting an indication of said response back to said debug program; and
- (e) repeating (b), (c) and (d) multiple times so as to run said application program through a diagnostic sequence.

10. The apparatus according to claim 9, wherein said diagnostic sequence is provided to said debug program by a user of said server computer.

11. The apparatus according to claim 9, wherein said diagnostic sequence is preprogrammed into said debug program.

12. The apparatus according to claim 11, wherein the running of said preprogrammed diagnostic sequence is paused by a user of said server computer and control of said debug program is transferred to said user of said server computer, and said interface program is further for, after (e):

- (f) receiving a request initiated by said user of said server computer;

(g) causing said application program to respond to said request;

(h) generating a graphics file including pixel information for a graphics image displayed on a display screen of said client computer; and

(i) transmitting said graphics file to said server computer so that said graphics image is displayed on a display screen of said server computer.

13. The apparatus according to claim 12, wherein said program is further for repeating (f) through (i) multiple times so as to allow said user of said server computer to remotely debug said application program.

14. A method implemented in a client computer for remotely debugging an application program over the Internet, comprising:

(a) establishing a connection between a client computer and a server computer over the Internet;

(b) receiving a request from a debug program of said server computer;

(c) causing an application program of said client computer to respond to said request;

(d) generating a graphics file including pixel information for a graphics image displayed on a display screen of said client computer; and

(e) transmitting said graphics file to said server computer so that said graphics image is displayable on a display screen of said server computer.

15. The method according to claim 14, wherein said graphics file is in a GIF, JPEG or TIF graphics file format.

16. The method according to claim 14, further comprising detecting a debug request initiated by a user of said client computer, prior to (a).

17. The method according to claim 14, further comprising repeating (b) through (e) multiple times so as to allow a user of said server computer to remotely debug said application program.

18. The method according to claim 14, further comprising transmitting identifications of said application program and said client computer to said server computer, after (a).

19. The method according to claim 14, further comprising transmitting a user identification and a password provided by a user of said client computer to said server computer, after (a).

20. An apparatus for remotely debugging an application program over the Internet, comprising a client computer including an interface program for:

- (a) establishing a connection between a client computer and a server computer over the Internet;
- (b) receiving a request from a debug program of said server computer;
- (c) causing an application program of said client computer to respond to said request;

(d) generating a graphics file including pixel information for a graphics image displayed on a display screen of said client computer; and

(e) transmitting said graphics file to said server computer so that said graphics image is displayed on a display screen of said server computer.

21. The apparatus according to claim 20, wherein said program is further for detecting a debug request initiated by a user of said client computer, prior to (a).

22. A method implemented in a server computer for remotely debugging an application program over the Internet, comprising:

(a) receiving a request from a client computer over the Internet to debug an application program of said client computer;

(b) transmitting back to said client computer a request for said application program to take an action;

(c) receiving an indication of a response of said application program action back from said client computer; and

(d) repeating (b) and (c) multiple times so as to run said application program through a diagnostic sequence.

23. The method according to claim 22, wherein said diagnostic sequence is provided to a debug program of a server computer by a user of said server computer.

24. The method according to claim 22, wherein said diagnostic sequence is preprogrammed into a debug program of a server computer.

25. The method according to claim 22, further comprising, prior to (b):

(a1) receiving an identification of said application program from said client computer; and

(a2) checking said application program identification against an application program identification list to confirm that a contractual obligation exists to debug said application program.

26. The method according to claim 22, further comprising, prior to (b):

(a3) receiving an identification of said client computer from said client computer; and

(a4) confirming that said client computer is authorized to run said application program by comparing said client computer identification against an authorized client computer identification.

27. The method according to claim 24, wherein the running of said preprogrammed diagnostic sequence is paused by a user of said server computer and control of said debug program is transferred to said user of said server computer, and further comprising:

(e) transmitting to said client computer a request for said application program to take an action;

(f) receiving a graphics file including pixel information for a graphics image displayed on a display screen of said client computer in response to said action;

(g) displaying said graphics image on a display screen of said server computer; and

(h) repeating (e) through (g) multiple times so as to allow said user of said server computer to interactively debug said application program by transmitting requests for said application program to take certain actions in consideration of graphics images defined in graphics files received from said client computer in response to prior such requests.

28. An apparatus for remotely debugging an application program over the Internet, comprising a server computer including a debug program for:

(a) receiving a request from a client computer over the Internet to debug an application program of said client computer;

(b) transmitting back to said client computer a request for said application program to take an action;

(c) receiving an indication of a response of said application program action back from said client computer; and

(d) repeating (b) and (c) multiple times so as to run said application program through a diagnostic sequence.

29. The apparatus according to claim 28, wherein said diagnostic sequence is provided to a debug program of a server computer by a user of said server computer.

30. The apparatus according to claim 28, wherein said diagnostic sequence is preprogrammed into a debug program of a server computer.

31. The apparatus according to claim 30, wherein the running of said preprogrammed diagnostic sequence is paused by a user of said server computer and control of said debug program is transferred to said user of said server computer, and further comprising:

(e) transmitting to said client computer a request for said application program to take an action;

(f) receiving a graphics file including pixel information for a graphics image displayed on a display screen of said client computer in response to said action;

(g) displaying said graphics image on a display screen of said server computer; and

(h) repeating (e) through (g) multiple times so as to allow said user of said server computer to interactively debug said application program by transmitting requests for said application program to take certain actions in consideration of graphics images defined in graphics files received from said client computer in response to prior such requests.

32. A method implemented in a server computer for remotely debugging an application program over the Internet, comprising:

(a) receiving a request from a client computer over the Internet to debug an application program of said client computer;

(b) transmitting back to said client computer a request for said application program to take an action;

(c) receiving a graphics file including pixel information for a graphics image displayed on a display screen of said client computer in response to said action;

(d) displaying said graphics image on a display screen of a server computer; and

(e) repeating (b) through (d) multiple times so as to allow a user of said server computer to interactively debug said application program by transmitting requests for said application program to take certain actions in consideration of graphics images defined in graphics files received from said client computer in response to prior such requests.

33. The method according to claim 32, wherein said graphics file is in a GIF, JPEG or TIF graphics file format.

34. The method according to claim 32, further comprising, prior to (b):

(a1) receiving an identification of said application program from said client computer; and

(a2) checking said application program identification against an application program identification list to confirm that a contractual obligation exists to debug said application program.

35. The method according to claim 34, further comprising, prior to (b):

(a3) receiving an identification of said client computer from said client computer; and

(a4) confirming that said client computer is authorized to run said application program by comparing said client computer identification against an authorized client computer identification.

36. The method according to claim 32, further comprising, prior to (b):

(a5) receiving a user identification and user password from said client computer; and

(a6) verifying that said user identification and said user password are valid.

37. An apparatus for remotely debugging an application program over the Internet, comprising a server computer including a debug program for:

(a) receiving a request from a client computer over the Internet to debug an application program of said client computer;

(b) transmitting back to said client computer a request for said application program to take an action;

(c) receiving a graphics file including pixel information for a graphics image displayed on a display screen of said client computer in response to said action;

(d) displaying said graphics image on a display screen of a server computer; and

(e) repeating (b) through (d) multiple times so as to allow a user of said server computer to interactively debug said application program by transmitting requests for said application program to take certain actions in consideration of graphics images defined in graphics files received from said client computer in response to prior such requests.

ABSTRACT OF THE DISCLOSURE

A method and apparatus for remotely debugging an application program over the Internet is described. A software vendor provides an interface program along with an application program to a customer. The interface program detects a debug request initiated by a customer operator on a customer computer, and establishes a connection with a debug program on a vendor computer over the Internet. The interface program takes over control of the application program from the customer operator, and takes various actions including running a diagnostic sequence received from the debug program on the application program to debug the application program.

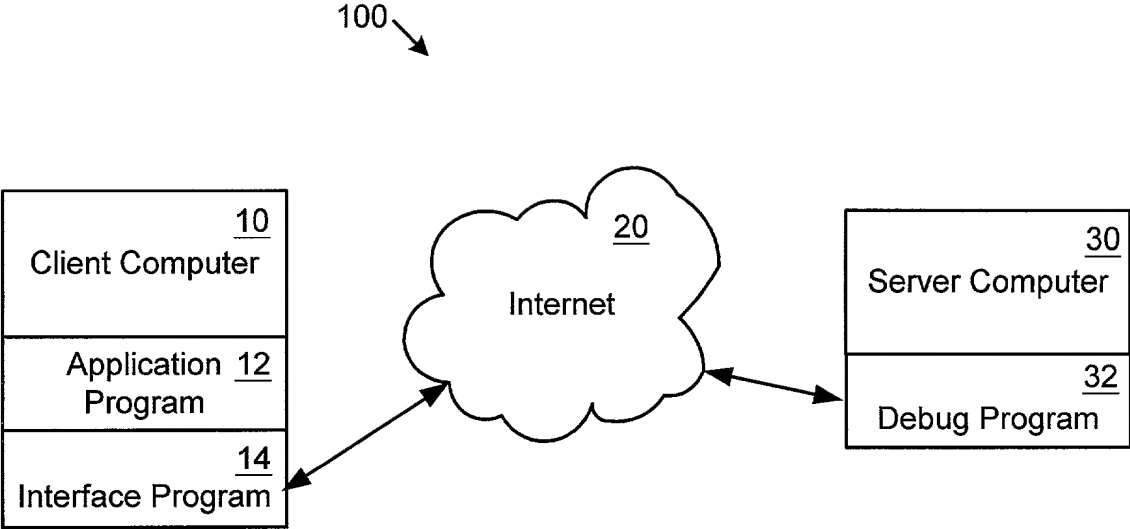


FIG.1

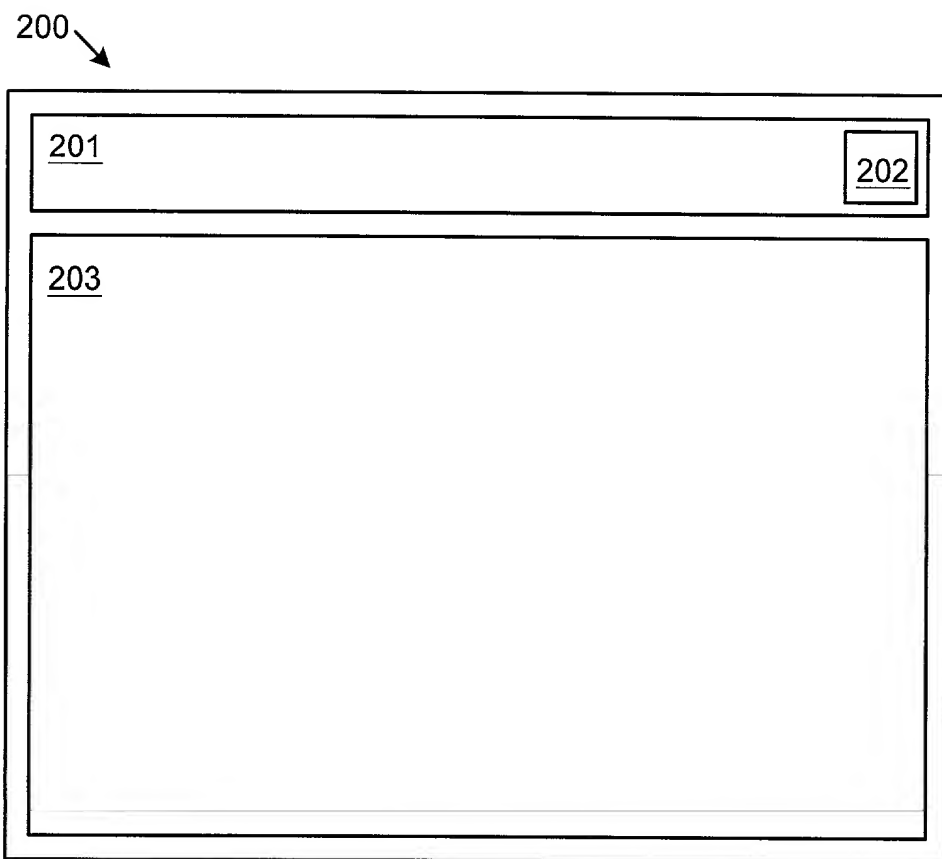


FIG.2

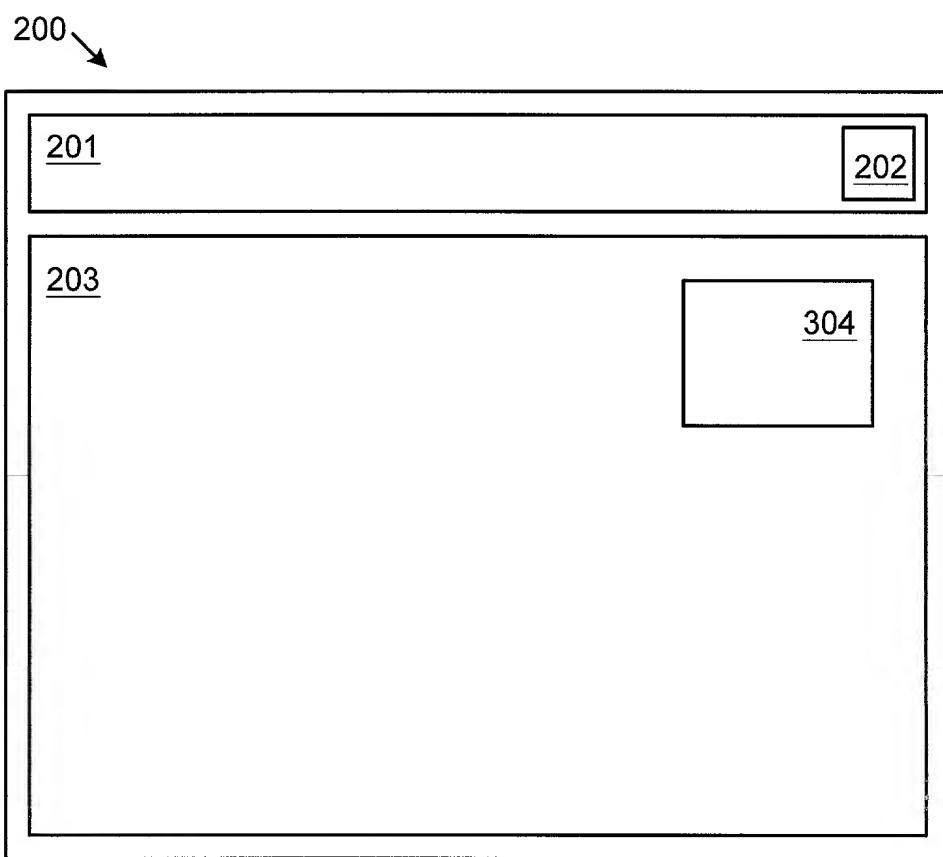


FIG.3

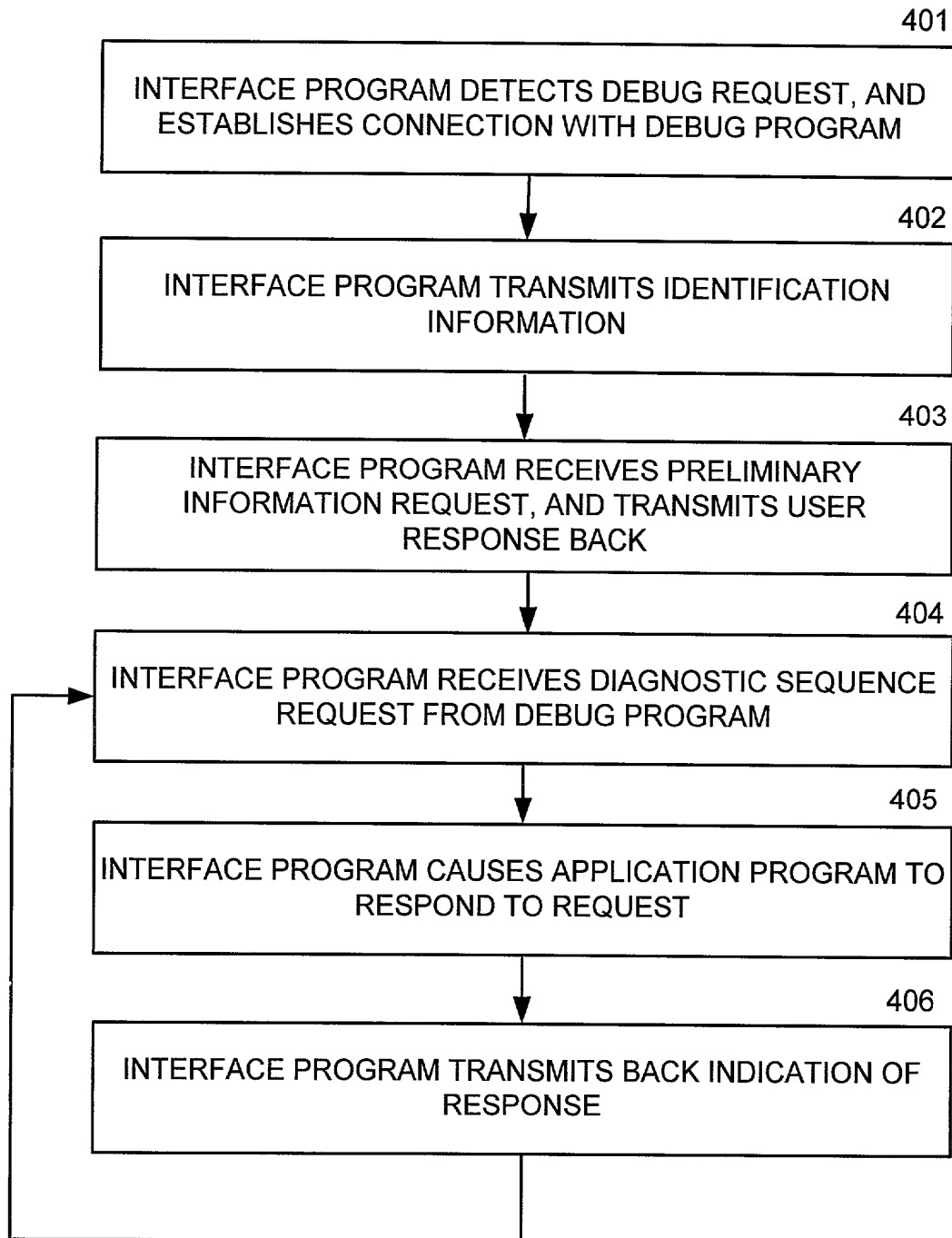


FIG.4

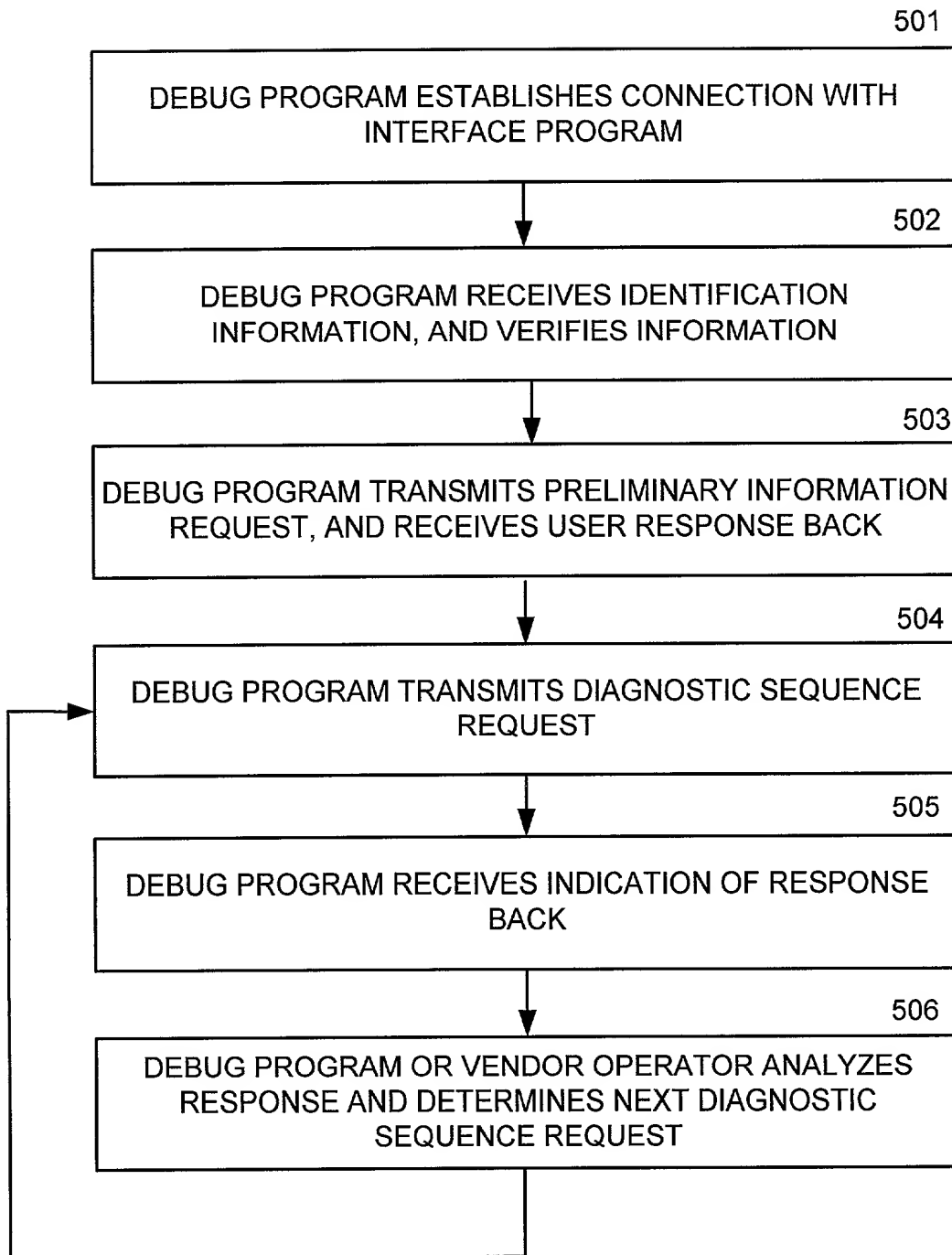


FIG.5

Docket No.
CREO.005US0

Declaration and Power of Attorney For Patent Application

English Language Declaration

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name,

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

METHOD AND APPARATUS FOR REMOTELY DEBUGGING AN APPLICATION PROGRAM OVER THE INTERNET

the specification of which

(check one)

☒ is attached hereto.

☐ was filed on _____ as United States Application No. or PCT International Application Number _____ and was amended on _____

(if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119(a)-(d) or Section 365(b) of any foreign application(s) for patent or inventor's certificate, or Section 365(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate or PCT International application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application(s)

Priority Not Claimed

_____ (Number)	_____ (Country)	_____ (Day/Month/Year Filed)	<input type="checkbox"/>
_____ (Number)	_____ (Country)	_____ (Day/Month/Year Filed)	<input type="checkbox"/>
_____ (Number)	_____ (Country)	_____ (Day/Month/Year Filed)	<input type="checkbox"/>

I hereby claim the benefit under 35 U.S.C. Section 119(e) of any United States provisional application(s) listed below:

(Application Serial No.)

(Filing Date)

(Application Serial No.)

(Filing Date)

(Application Serial No.)

(Filing Date)

I hereby claim the benefit under 35 U. S. C. Section 120 of any United States application(s), or Section 365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of 35 U.S.C. Section 112, I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in Title 37, C. F. R., Section 1.56 which became available between the filing date of the prior application and the national or PCT International filing date of this application:

(Application Serial No.)

(Filing Date)

(Status)
(patented, pending, abandoned)

(Application Serial No.)

(Filing Date)

(Status)
(patented, pending, abandoned)

(Application Serial No.)

(Filing Date)

(Status)
(patented, pending, abandoned)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

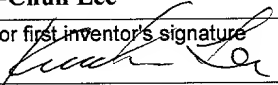
POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. *(list name and registration number)*


Victor H. Okumoto

35,973

Send Correspondence to: Victor H. Okumoto
P.O. Box 6120
Fremont, CA 94538

Direct Telephone Calls to: *(name and telephone number)*
Victor H. Okumoto (510) 792-1112

Full name of sole or first inventor	Kuo-Chun Lee	
Sole or first inventor's signature		Date 9/19/2000
Residence	139 Buck Court, Fremont, CA 94539	
Citizenship	Taiwan	
Post Office Address		

Full name of second inventor, if any	Tsun-Yen (Eric) Chen	
Second inventor's signature		Date Sep. 19, 00
Residence	43519 Puesta Del Sol, Fremont, CA 94539	
Citizenship	Taiwan	
Post Office Address		